**暨南大学本科实验报告专用纸**

课程名称 C++程序设计 成绩评定

实验项目名称 unique\_ptr编程 指导教师 王勇杰

实验项目编号 5 实验项目类型 编程型 实验地点 C105

学生姓名 李炜鹏 学号 2017052544

学院 智能科学与工程学院 系 计算机 专业 信息安全

实验时间2019年12月10日 午～12月13日 午 温度 ℃湿度

1. **实验目的**
2. 了解智能指针的两大类：shared\_ptr、unique\_ptr的概念及异同
3. 了解并熟悉两类智能指针的基本用法
4. 实验要求：阅读课本第12章，调试运行“实验五 unique\_ptr编程”文件夹中的源代码。
5. 实验报告要求：
   1. 实验目的
   2. 实验原理
   3. 注释代码中的每条语句
   4. 实验结果的截图
   5. 实验评估
6. **实验原理**
7. shared\_ptr：
   1. 初始化：最安全的方法是使用make\_shared函数；也可以使用new，但由于接受指针参数的智能指针构造函数是explicit的，因此不能够将一个内置指针隐式转换为一个智能指针，必须使用直接初始化形式来初始化一个智能指针：

shared\_ptr<int> p1 = make\_shared<int>(1024); //正确，使用make\_shared函数

shared\_ptr<int> p2 = new int(1024); //错误，不能将内置指针隐式转换为智能指针

shared\_ptr<int> p2(new int(1024)); //正确，直接初始化

* 1. 可以将智能指针绑定到一个指向其他类型的资源的指针上，但必须提供自己的操作来替代delete（参考教材C++ Primer（第5版）P416）

1. unique\_ptr：

与shared\_ptr不同的是：在某个时刻只能有一个unique\_ptr指向给定对象；unique\_ptr支持索引而shared\_ptr不支持索引；unique\_ptr可以用release函数释放对象

1. **主要仪器设备**

**仪器：**计算机

**实验环境：**Visual Studio 2017或Dev-C++

1. **源程序**

**UP.cpp**

#include <iostream>

#include <memory>

using namespace std;

using std::shared\_ptr;

using std::unique\_ptr;

void deleter(int \*p) { delete[] p; }//删除器

int main()

{

//在pre-c++11库中不支持unique\_ptr

unique\_ptr<int[]> up(new int[10]);

for (size\_t i = 0; i != 10; ++i)

{

up[i] = i;

cout << up[i] << endl;

}

up.release(); //自动使用delete[]来销毁它的指针

cout << endl;

// 要使用shared\_ptr我们必须提供一个删除器

shared\_ptr<int> sp(new int[10], deleter);//定义一个共享智能指针，指向一个含10个int型数据的数组，并且指定销毁时使用的删除器

// shared\_ptr没有下标操作符

// 也不支持指针运算

for (size\_t i = 0; i != 10; ++i)

{

\*(sp.get() + i) = i; //用get函数来取得共享智能指针的地址

cout << \*(sp.get()+i) << endl;

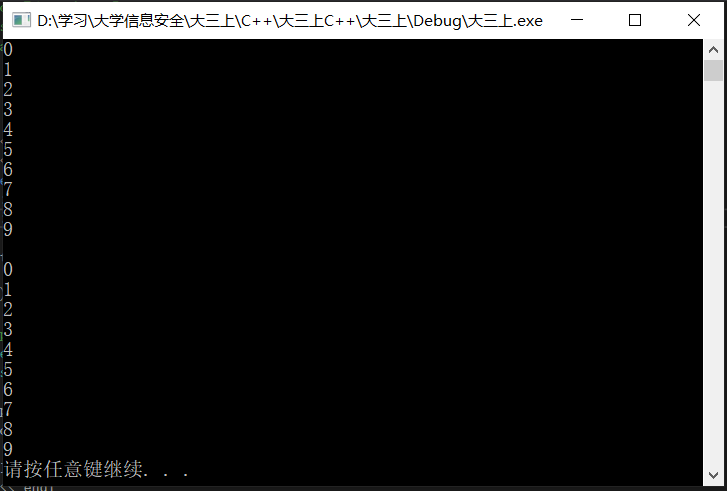
}

sp.reset();//这里使用我们定义的删除器函数来销毁int数组对象

system("pause");

return 0;

}



分别使用unique\_ptr和shared\_ptr来动态分配int型数组内存，unique\_ptr支持下标运算，shared\_ptr不支持下标运算，本实验分别使用了这两者的不同用法，运行效果符合预期。

1. **实验评估**

通过本实验，我初步了解到了两类智能指针：shared\_ptr和unique\_ptr，认识了其定义、初始化、引用以及销毁的方法，了解到了两者之间的异同。还认识到了普通的内置指针与智能指针之间的联系与区别（普通指针直接使用new和delete关键字来分配与销毁对象，而智能指针能够自动销毁对象及对应的内存空间）。